BAYESIAN TV SHOW RECOMMENDER

A PATENT APPLICATION BY:

INVENTOR:

J. DAVID SCHAFFER

DRAFTING ATTORNEY

ANNE E. BARSCHALL

REG. NO. 31,089

I. BACKGROUND OF THE INVENTION

A. Field of the Invention

The invention relates to recommending television shows based on a user profile.

B. Related Art

`U.S. Pat. No. 5,758,259 shows a method for identifying a preferred television program based on a "correlation" between the program and predetermined characteristics of a user profile. The term "correlation" as used in the patent does not appear to relate to the mathematical concept of correlation, but rather is a very simple algorithm for assessing some similarity between a profile and a program.

II. SUMMARY OF THE INVENTION

It is an object of the invention to improve techniques of automatic program recommendation.

This object is achieved by using a probabilistic calculation, based on a viewer profile. The probabilistic calculation is preferably based on Bayesian classifier theory.

The object is further achieved by maintaining a local record

of a viewer history. The local record is preferably incrementally updatable. The local record is advantageous for privacy reasons, and can be contrasted with methods such as collaborative filtering, which would require viewer history information to be uploaded to a central location. The use of incremental updates is advantageous in minimizing storage requirements.

It is a still further object of the invention to improve the classical Bayesian classifier technique.

In one embodiment, this object is achieved by noise filtering.

In another embodiment, this object is achieved by applying a modified Bayesian classifier technique to non-independent feature values.

Further objects and advantages of the invention will be described in the following.

Bayesian classifiers are discussed in general in the text book Duda & Hart, Pattern Recognition and Scene Analysis (John Wiley & Sons 1973). An application of Bayesian classifiers to document retrieval is discussed in D. Billsus & M. Pazzani, Learning Probabilistic User Models", ***http://www.dkfi.uin-sb.de/~bauer/um-ws/Final-Versions/Billsus/ProbUserModels.html***

III. BRIEF DESCRIPTION OF THE DRAWING

The invention will now be described by way of non-limiting example with reference to the following drawings.

Fig. 0 shows a system on which the invention may be used.

Fig. 1 shows major elements of an adaptive recommender.

Fig. 2 shows pseudo code for a viewing history generator.

Fig. 3 shows a table of key fields.

Fig. 4 shows a fragment of a viewer profile.

Fig. 5a shows a prior probability calculation.

Fig. 5b shows a conditional probability calculation.

Fig. 5c shows a posterior probability calculation.

IV. DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Fig 0 illustrates hardware for implementing the invention. The hardware will typically have a display 1; some type of processor 2; some type of user entry device 4 connected to the processor via some type of connection 3; and some type of link 5 for receiving data, such as television programming or Electronic Programming Guide ("EPG") data. The display 1 will commonly be a television screen, but could be any other type of display device. The processor 2 may be a set top box, a PC, or any other type of data processing device, so long as it has sufficient processing power. The user entry device 4 may be a remote and the

connection 3 may be an infrared connection. If the processor is a PC, the user entry device will commonly be at least plural, e.g. a keyboard and mouse. The user entry device may also be touch sensitivity on the display. The connection 5 to the outside world could be an antenna, cable, a phone line to the internet, a network connection, or any other data link. Equally well, connection 5 could connect to a memory device or several memory devices.

Fig. 1 illustrates major elements of an embodiment of an adaptive recommender. These elements preferably reside as software and data in a medium 110 readable by a data processing device such as CPU 2. The elements include a viewing history data structure 101 that gives input to profiler software 102. The profiler software in turn produces the viewer profile 103. The terms "user profile" and "viewer profile" shall be used interchangeably herein. The viewer profile serves as input to recommender software 104. The recommender software also uses, as input, the EPG data structure 105, that contains features describing each show such as title, channel, start time and the like. An output of the recommender 104 appears on a user interface 106 where a user can interact with it.

This viewer history data structure includes selected records from the EPG database. The EPG databases are commercially

available, for instance from Tribune Media Services. Those of ordinary skill in the art may devise other formats, possibly with finer shades of description. The selected records minimally correspond to TV shows watched by the viewer. It is assumed that these records have been deposited in the viewing history by software that is part of the user interface and knows what shows the viewer has viewed. Preferably, the software would allow recording of a user watching more than one show in a given time interval, as users often do switch back and forth during commercials and so forth. Preferably, also, the software records a program as watched; and whether a show was watched or whether it was taped for later viewing.

The preferred viewing history format assumes the presence of both positive and negative records in the viewing history. This is needed because the goal is to learn to differentiate between the features of shows that are liked and those not liked. Fig. 2 shows pseudo code for collecting the viewing history.

Let the notation *C+* denote the set of positive (i.e., watched) shows and *C-* denote the negative (i.e., not watched) shows.

The viewer profile includes a number of feature value counts. These counts will be incremented whenever new entries are deposited in the viewer history. Usually, each program will

have several feature values. Accordingly, the deposit of a program in the viewer history will cause the update of counts associated with all feature values associated with that program.

The incremental updatability of this type of profile is advantageous because it allows for ongoing adaptation of the viewer profile without a large amount of storage or computing effort being required.

In addition to the count of the number of positive and negative entries (k(C+), k(C-)), a count of occurrences of individual features will also be kept among the positive and negative examples (k(fi|C+), k(fi|C-)) where fi denotes feature i and k(fi|C+) denotes the number of shows in set C+ that possess feature fi. The feature set will include entries in the EPG records extracted from selected key fields, an example of which is shown in Table 1 which is Fig. 3.

A partial example of an embodiment of such counts is presented in table 2, shown in Fig. 4 to illustrate the idea. The list is presented in the Figure in six columns to save space, but in fact the list has only three columns, with the later part of the table being presented next to the earlier part. Each row of the column has four pieces of data: a feature type and a feature value in the first column, a positive count in the second column, and a negative count in the third column. The positive count

indicates the number of times a program having that feature value has been watched. The negative count indicates the number of times a program having that feature value has not been watched.

A television program schedule normally includes several, if not many, programs for every time slot in every day. Normally, the user will only watch one or two of the programs in any given slot. If the viewer profile contains a list of ALL the programs not watched, the number of programs not watched will far exceed the number of programs watched. It may be desirable to create a method for sampling the programs not watched. For instance, as the processor assembles the viewer profile, the processor may chose a single not watched program at random from the weekly schedule as a companion for each watched program, as suggested in the pseudocode of Fig. 2. This design will attempt to keep the number of positive and negative entries in the viewing history about equal so as not to unbalance the Bayesian prior probability estimates, discussed below.

It is not generally desirable to choose a companion program from the same time slot as a watched program. Experiment has shown that the combined time and day feature value is typically the strongest or one of the strongest predictors of whether a particular program will be preferred. Thus another program at the same time as the watched program may well be a second or

third choice program, while a program at a totally different time may be very undesirable. Accordingly, it is preferred to choose the companion program at random from the program schedule of the entire week that includes the watched program.

Since time and day feature values for a program are often so important in determining whether a program will be of interest to a user, it is typically undesirable to consider two programs of identical content to be the same if they are shown on different days and/or at different times. In other words a particular episode of a series may be strongly preferred if it is shown at 8 p.m. on Tuesday, while the same episode of the same series may be completely undesirable if it is shown at 10 a.m. on Monday. Thus the episode at 10 a.m. should be considered a different program from the episode at 8 p.m., even though the content of the two are identical.

As more and more shows are viewed, the length of the profile will tend to grow larger and larger. To combat this, and to keep the focus on features that are effective discriminators, the following are recommended:

- periodic reviews of the features in the viewer profile, and
- removal of words that appear to be frequent and not very discriminating.

In general, those of simple tastes, e.g. those who only like to watch football, will be fairly easy to recommend for after taking of a viewer history for a relatively short time. For those of more complex preferences, it will take longer for the viewer history to be sufficiently meaningful to make good recommendations. These latter people, however, are those who are probably most in need of a recommendation.

In the final analysis, viewer histories will always be ambiguous. Recommendations of shows based on such histories will always contain a margin for error. The recommendations can at best be said to have some probability of being correct. Therefore probabilistic calculations are useful in analyzing viewer profile data to make recommendations.

The preferred embodiment of the recommender uses a simple Bayesian classifier using prior and conditional probability estimates derived from the viewer profile. How recommendations are shown to viewers is not defined here, yet it will be assumed that one can capture the viewer's response to them, at least observing whether or not they were watched.

Below will be discussed a 2-class Bayesian decision model. The two classes of TV shows of interest are:

C1 - shows that interest the viewer

C2 - shows that do not interest the viewer

Other classes might be used showing more shades of interest or lack thereof.

In contrast with the classes of interest listed above, viewing history contains information only on the classes:

C+ - shows the viewer watched

C- - shows the viewer did not watch

Determining which shows a user watched or did not watch is outside of the scope of this application. The user might enter a manual log of which shows s/he watched. Alternatively, hardware might record the user's watching behavior. Those of ordinary skill in the art might devise numerous techniques for this. It should be possible to consider shows as watched even if they are watched only for a short time, as a user may be switching back and forth between several shows, trying to keep track of all of

them.

Inferences may be made about classes *C1* and *C2* based on observations, but these inferences will always contain an element of uncertainty. The Bayesian model will compute the prior probabilities *P(C+)* and *P(C-)* directly from the counts in the viewer profile in accordance with Fig. 5a. In other words, the assumption will be that shows not watched are generally those the viewer is not interested in, and that the shows watched are the ones that the viewer is interested in.

The conditional probabilities, that a given feature, *fi*, will be present if a show is in class *C+* or *C-*, are then computed in accordance with Fig. 5b. These calculations can be performed once a day during times that the TV is not being viewed and stored in the viewer profile.

Recommendations for upcoming shows can be computed by estimating the posterior probabilities, i.e. the probability that a show is in class *C+* and *C-* given its features. Let ***x*** be a binary vector *(x1, x2, …,xi, …, xn)* where *i* indexes over the features in the viewer profile and where *xi* = 1 if feature *fi* is present in the show being considered for recommendation and 0 if not. For the exclusive features, like day, time, and station, where every show must have one and only one value, the index *i* will be taken to indicate the value present in the show being

considered provided that this value is also present in the profile. Otherwise, novel exclusive features will not enter into the calculations. For non-exclusive features, the index *i* will range over all values present in the profile; non-exclusive features novel to the considered show will not contribute to the calculations. The posterior probabilities are estimated in accordance with Fig. 5c

With these estimates in hand, a show will generally be recommended if $P(C+|\mathbf{x}) > P(C-|\mathbf{x})$ and the "strength" of the recommendation will be proportional to $P(C+|\mathbf{x}) - P(C-|\mathbf{x})$. One potential problem with this scheme is that some conditional probabilities are likely to be zero. Any zero in a chain multiplication will reduce the result to zero so some means for eliminating zeros is needed. The Billsus and Pazzani article referenced above presents a couple of schemes, including simply inserting a small constant for any zeros that occur.

One method for dealing with zeroes in the conditional probability multiplication chain would be as follows. One can choose a heuristic of 1000. If the number of shows in the viewing history is less than 1000, then the value of 1/1000 can be substituted for zero. If the number of shows in the viewing history is greater than 1000, the correction can be

$$\frac{k_i + \ \ +1}{k+ \ \ +2}$$

Where

k_i+ is the number of watched shows having feature i

k+ is the total number of watched shows.

This is what is called the Laplace correction in the Billsus and Pazzani article. This Laplace correction must also be done for the not watched shows.

Alternative schemes may be devised by those of ordinary skill in the art.

Classical Bayesian theory would require the use of all accumulated elements of the list of Fig. 4 in making a recommendation. Nevertheless, in some instances it may be useful to use a noise cutoff, eliminating features from consideration if insufficient data about them appears in the list. For instance if a particularly feature did not appear in more than some given percentage of shows considered, whether in negative or in positive count, it might be ignored in determining which recommendation to make. Experimentally it was found that a cutoff of 5% was far too large.

Rather than use a percentage, one embodiment of the noise cutoff would use the viewer profile itself to determine the cutoff. This embodiment would first take a subset, or sub-list, of the viewer profile relating to particular feature types. For

instance, a sub-list might advantageously comprise all of the elements of the viewer profile relating to the feature types: time of day and day of the week. Alternatively, in another example, the sub-list might advantageously comprise all of the elements of the viewer profile relating to channel number. Generally the feature type or types chosen should be independent feature types, in other words feature types which do not require another feature type to be meaningful.

The sub-list is then sorted by negative count, i.e. by number of shows having a particular feature value and not being watched. The highest negative count in this sorted list can be viewed as the noise level. In other words, since, in the preferred embodiment, the "not watched" shows are chosen at random from the week's program schedule, counts as large as the noise level can occur by chance, and therefore should be ignored.

Thus any feature having both a positive and a negative count at or below the noise level need not be considered in the Bayesian calculation in making a recommendation. This example of noise level thresholding has used a particular feature, e.g. day/time as one for determining noise cutoff. In general, any feature that is uniformly randomly sampled by the negative example sampling procedure may be chosen by those of ordinary skill in the art for the calculation of the noise threshold.

The calculations of Figs. 5a-c are advantageous in that they require fairly low computing power to complete and are therefore readily adaptable to modest hardware such as would be found in a set top box.

"Surprise Me" Feature

Recommendations according to the above-described scheme will be programs having a preponderance of features that are present in shows that have been watched. The viewer profiles accumulated will not yield any meaningful recommendations with respect to shows having few features in common with those features that are in the watched and not watched shows and register above the noise level. Accordingly, optionally, the recommender may occasionally recommend shows at random, in a "surprise me" feature. The surprise me feature would recommend programs with relatively few features in common with watched and not watched shows, to the extent that such features register above the noise level.

Using the user profile in other domains

Once a user profile is developed, the recommendation techniques of the invention might be used to recommend other types of items such as movies, books, audio recordings, or even promotional materials such as tee shirts or posters.

Non-independence of Features

The classical assumption in the domain of Bayesian classifier theory is that all features are independent. Therefore, if a features is, say, often present in positive shows, but is missing from a show being considered for recommendation, the fact should count against the show. However, this may yield undesirable results for the current application.

For example, let us assume that there are five day/time slots indicated in the user profile as being most watched. Let us assume further that a particular show being evaluated falls within one of those five slots. The calculation of Fig. 5c would then give rise to an increase in probability for the day/time slot that matches and a decrease for the four day/time slots that do not match. Intuitively, it appears that the latter decrease is not reasonably related to an accurate determination of probability for the show in question. The different values of day/time are not independent -- as every show has one and only one value, so the values a show does not have should not count against it.

To remedy this deficiency in the classical Bayesian approach, it is proposed to designate features into two types: set 1 and set 2. If a feature is designated set 1, the Bayesian

calculation will ignore any non-matching values of the feature. If the feature is designated set 2, then the normal Bayesian calculation, per Fig. 5c, will be done.

Normally in a television application set 1 would include day/time; station; and title. Some features which have values only for a few shows, e.g. critic ratings, should also be set 1, because too many shows would be non-matching merely because critics tend to rate only a tiny percentage of shows.

Set 2, for television shows, would normally include all features that can have several values per show, such as actor.

From reading the present disclosure, other modifications will be apparent to persons skilled in the art. Such modifications may involve other features which are already known in the design, manufacture and use of television interfaces and which may be used instead of or in addition to features already described herein. Although claims have been formulated in this application to particular combinations of features, it should be understood that the scope of the disclosure of the present application also includes any novel feature or novel combination of features disclosed herein either explicitly or implicitly or any generalization thereof, whether or not it mitigates any or all of the same technical problems as does the present invention. The applicants hereby give notice that new claims may be

formulated to such features during the prosecution of the present application or any further application derived therefrom.

The word "comprising", "comprise", or "comprises" as used herein should not be viewed as excluding additional elements. The singular article "a" or "an" as used herein should not be viewed as excluding a plurality of elements.